2016

3D FPGA Cell Matrix by Self-assembly

Jeffrey Udall
Portland State University

3D FPGA Cell Matrix by Self-assembly

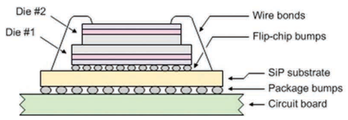
Jeffrey Udall - judall@pdx.edu

Portland State University - Maseeh College of Engineering and Computer Sciences

Maseeh College of Engineering and Computer Science
PORTLAND STATE UNIVERSITY

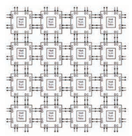
Introduction

Three-dimensional architectures, currently implemented via wafer stacking, are one solution to the physical size limitations encountered in miniaturizing two-dimensional transistors.

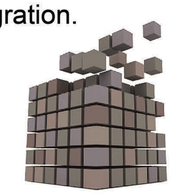


2D circuits in a 3D stacked arragement

More significant gains in packing and speed can be achieved by CMOS components with truly integrated 3D cellular architectures such as the Cell Matrix, a self-configurable defect- and fault-tolerant reconfigurable array of cells, ideally suited for ultra large-scale integration.



Interconnected cells (2D)



3D interconnected cells

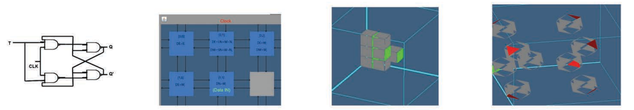
Methods

For this project, we worked to expand the Cell Matrix models and tools from 3D cubes to a 3D truncated octahedron geometry.

Truncated Octahedron

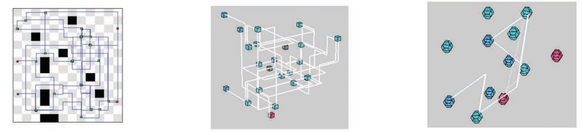
Interconnected TO cells

Using a Java simulator we can implement circuit designs using the Cell Matrix architecture.



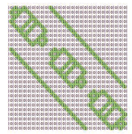
Flip flop circuit implemented in 2D, 3D cube and 3D TO cells (Java)

Using MATLAB place and route (PAR) program the shortest average path can be simulated and compared.

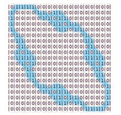


Flip flop circuit implemented in 2D, 3D cube and 3D TO cells (MATLAB)

Because of issues with using the MATLAB PAR we also created a program in MATLAB to calculate the total connections in matrices with different topologies.



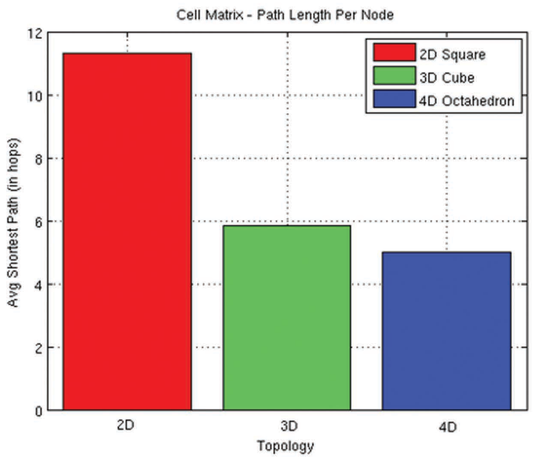
3D cube connections
27 nodes - 54 connections



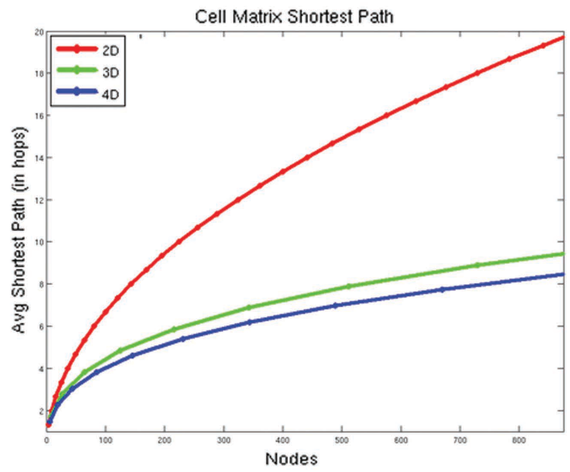
3D TO connections
19 nodes - 40 connections

Results

Because the efforts to implement the truncated octahedron topologies in MATLAB PAR were unsuccessful the resulting data is only from the shortest path calculations between cells in different topologies.



4D (TO) cell arrays have a shorter average path between nodes in a cell matrix array



The average shortest path per node decreases faster with 4D (TO) cells than 2D and (to a lesser extent) 3D cells

Comparing the path lengths connecting the nodes versus the number of nodes in the matrices the truncated octahedron (4D) matrices have a shorter total path length per number of nodes.

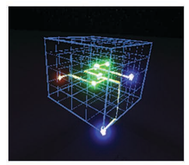
Conclusions

The shorter average paths in the truncated octahedron (4D) arrays indicate that a speed increase could be achieved by using a truncated octahedron topology over cube or square topologies.

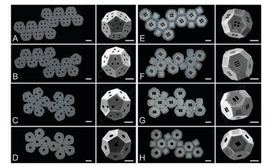
Further research will be needed to conclude if the advantages in the truncated octahedron topology result in a greater overall computational advantage versus total cost.

Future Research

For future research we are looking into improving the shortest path program to place circuits in arrays to be closer to the MATLAB PAR. We are also looking into VR application to visualize 3D circuit placement as well as other topologies



3D circuit drawn in VR program (Tiltbrush)



Alternate 3D topologies to explore

Acknowledgments: The authors acknowledge the support of the **Semiconductor Research Corporation** (**SRC**) **Education Alliance** (*award # 2009-UR-2032G*) and of the **Maseeh College of Engineering and Computer Science** (**MCECS**) through the **Undergraduate Research and Mentoring Program** (**URMP**) also **Christof Teuscher**, Portland State University ECE Dept - **Nick Macias**, Cell Matrix Corp - **David Gracias**, Johns Hopkins University

References:

1. Durbeck, et al. (2001). The Cell Matrix: an architecture for nanocomputing. Nanotechnology, 12, 217–230. doi:10.1088/0957-4484/12/3/305
2. Macias, et al. (2013). A cellular architecture for self-assembled 3D computational devices. NANOARCH 2013, 116–121. doi:10.1109/NanoArch.2013.6623055
3. Yatsenko, D. (2003). Automated Placement and Routing of Cell Matrix Circuits. Utah State University.